The Stata Journal (*yyyy*) *vv*, Number *ii*, pp. 1–26

Analysing repeated measurements whilst accounting for derivative tracking, varying within-subject variance, and autocorrelation: the xtmixediou command

Rachael A. Hughes
School of Social and Community Medicine
University of Bristol
Bristol, United Kingdom
rachael.hughes@bristol.ac.uk

Michael G. Kenward
Luton, United Kingdom
michael.kenward@lshtm.ac.uk

Jonathan A.C. Sterne
School of Social and Community Medicine
University of Bristol
Bristol, United Kingdom
jonathan.sterne@bristol.ac.uk

Kate Tilling
School of Social and Community Medicine; MRC Integrative Epidemiology Unit
University of Bristol
Bristol, United Kingdom
kate.tilling@bristol.ac.uk

Abstract.

Linear mixed effects models are commonly used to model trajectories of repeated measures of biomarkers of disease. Taylor et al (Taylor, Cumberland, and Sy, 1994, J Am Stat Assoc 89: 727-736) proposed a linear mixed effects model with an added Integrated Ornstein Uhlenbeck (IOU) process (linear mixed effects IOU model). This allows for autocorrelation, changing within-subject variance, and the incorporation of derivative tracking; that is, how much a subject tends to maintain the same trajectory for extended periods of time. They argued that the covariance structure induced by the stochastic process in this model was interpretable and more biologically plausible than the standard linear mixed effects model. However, their model is rarely used, partly due to the lack of available software. We present a new Stata command, `xtmixediou`, that fits the linear mixed effects IOU model, and its special case the linear mixed effects Brownian Motion model. The model is fitted, to balanced and unbalanced data, using restricted maximum likelihood estimation, where the optimization algorithm is the Newton-Raphson, Fisher Scoring or Average Information algorithm, or any combination of these. To aid convergence the command allows the user to change the method for deriving the starting values for optimization, the optimization algorithm, and the parameterization of the IOU process. We also provide a `predict` command

 Published version available from http://www.stata-journal.com

to generate predictions under the model. We illustrate `xtmixediou` and `predict` with a simulated example of repeated biomarker measurements from HIV-positive patients.

Keywords: Published version available from http://www.stata-journal.com, xtmixediou, xtmixedioupredict, autocorrelation, derivative tracking, Integrated Ornstein Uhlenbeck process, repeated measures data, within-subject variability

1 Introduction

Linear mixed effects models, proposed by Laird and Ware (1982), are commonly used to model trajectories of repeated measures of biomarkers of disease; for example, trajectories of CD4 counts in HIV-positive patients (Boscardin et al. 1998) and trajectories of progesterone during a menstrual cycle (Sowers et al. 1998). In such settings the data are typically unbalanced: the number of measurements differs between subjects and the time interval between consecutive measurements differs within and between subjects. The variance of the biomarker may be nonstationary (vary over time) and, when measurements on the same subject are recorded close together in time, within-subject measurements may be serially correlated (also known as autocorrelation).

Taylor et al proposed a model where between-subject and within-subject variability are described by subject-level random effects, an Integrated Ornstein Uhlenbeck (IOU) stochastic process, and measurement errors. We shall refer to Taylor's model as the linear mixed effects IOU model, and a model without the IOU process (i.e., including only fixed-, subject-level random effects, and measurement errors) as a standard linear mixed effects model. The linear mixed effects IOU model estimates the degree of derivative tracking from the data; that is, how much a subject's measurements maintain the same trajectory over long periods. It covers a range of models from strong derivative tracking to no derivative tracking. Figure 1 shows predicted biomaker measurements for a subject generated under four linear mixed effects models with different degrees of derivative tracking. The model without an IOU process corresponds to a standard linear mixed effects model, which assumes strong derivative tracking (i.e., maintains the same trajectory throughout), and so a subject's predicted measurements identically track the parametric trajectory (a linear slope) given by the fixed and random effects. The remaining three models include an IOU process, where weaker degrees of derivative tracking correspond to greater departures in the path of the predicted measurements from the parametric trajectory. Taylor et al. argued that it is unlikely that a complex biomarker such as CD4 cell counts would maintain the same trajectory over long periods, and so the linear mixed effects IOU model was more biologically plausible than a standard linear mixed effects model. Unlike a standard linear mixed effects model, the linear mixed effects IOU model also allows for autocorrelation and non-constant within-subject variance. Based on a simulation study, Taylor and Law (1998) concluded that, when predicting future measurements in subjects, the linear mixed effects IOU model was more robust than a standard linear mixed effects model to incorrect specification of the true covariance structure of the data. Previously, the authors have evaluated the feasibility and practicality of estimating the linear mixed effects IOU model (Hughes et al.

2017). The model is rarely used in practice due to the lack of available software.

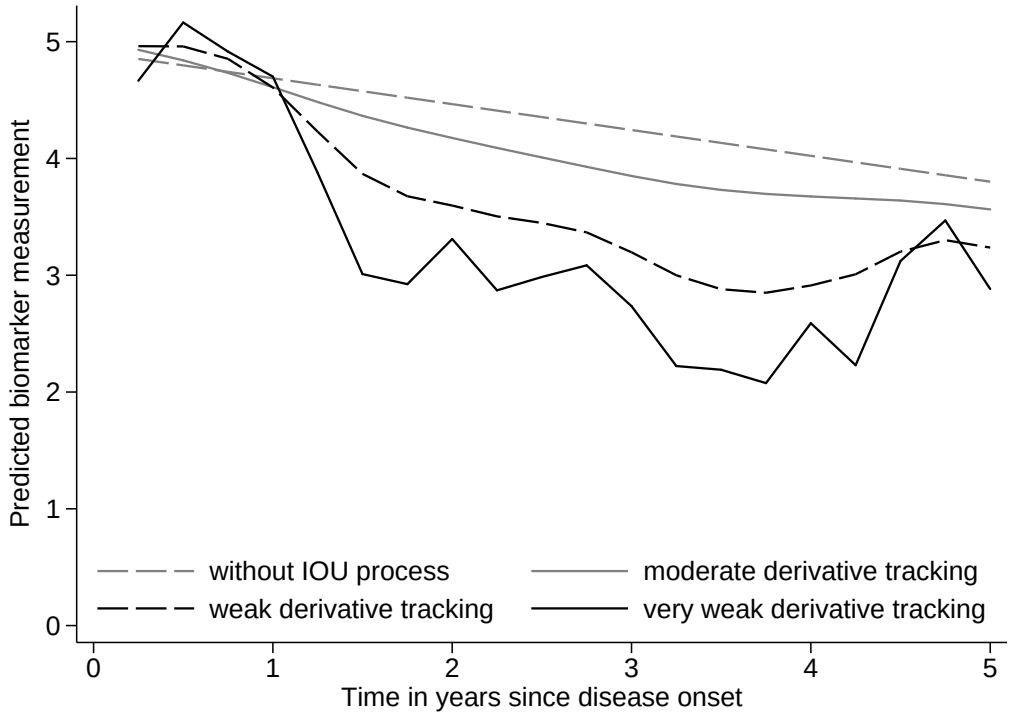

Figure 1: Different degrees of derivative tracking

This article describes `xtmixediou`, a new Stata command that fits the linear mixed effects IOU model and the corresponding `predict` command to generate predictions under this model. We illustrate the `xtmixediou` command using simulated data of repeated measurements of an immunologic marker (CD4 cell count) from HIV-positive subjects. We examine the variance structures of 6 different linear mixed effects models and compare the accuracy of predictions under these models.

2 The linear mixed effects Integrated Ornstein Uhlenbeck model

Consider a dataset of m subjects, where subject i has n_i repeated measurements $\mathbf{y_i} = \{y_{ij}\}$ recorded at timepoints $\mathbf{t_i} = \{t_{ij}\}$ $(i = 1, \ldots, m;\ j = 1, \ldots, n_i)$. For subject i, let $\mathbf{X_i} = \{X_{ij}\}$ denote the $n_i \times p$ design matrix associated with fixed effects β (population regression coefficients), $\mathbf{Z_i}$ the $n_i \times q$ design matrix associated with random effects $\mathbf{u_i}$ (subject-specific regression coefficients), $\mathbf{w_i} = \{\mathbf{w_{ij}}\}$ the n_i-vector of realized values of

the IOU stochastic process and $\mathbf{e_i}$ the n_i-vector of independent measurement errors. The random effects $\mathbf{u_i}$, IOU realizations $\mathbf{w_i}$, and measurement errors $\mathbf{e_i}$ are assumed to be mutually independent.

The linear mixed effects IOU model can be written as

$$\mathbf{y_i} = \mathbf{X_i}\beta + \mathbf{Z_i}\mathbf{u_i} + \mathbf{w_i} + \mathbf{e_i}, \tag{1}$$

where $\mathbf{u_i}, \mathbf{w_i}$, and $\mathbf{e_i}$ are independently and Normally distributed with zero means and covariances $\mathbf{G}$, $\mathbf{H_i}$, and $\sigma^2\mathbf{I_{n_i}}$ respectively. $\mathbf{G}$ is unstructured (i.e., variances and covariances distinctly estimated) and $\mathbf{H_i}$ is defined as follows, for $j_1, j_2 = 1, \ldots, n_i$,

$$\begin{aligned} H_i^{j_1 j_2} &= \frac{\tau^2}{2\alpha^3} \times [2\alpha\, min(t_{ij_1}, t_{ij_2}) \\ &\quad + \exp(-\alpha t_{ij_1}) + \exp(-\alpha t_{ij_2}) - 1 - \exp(-\alpha|t_{ij_1} - t_{ij_2}|)]. \end{aligned}$$

The IOU stochastic process is parameterized by α and τ. Taylor et al. (1994) state that α can be interpreted as a measure of the degree of derivative tracking, where a small value of α indicates strong derivative tracking. Parameter τ serves as a scaling parameter. As α tends to ∞ (derivative tracking becomes progressively weaker), and with ratio τ^2/α^2 held constant, $\mathbf{w_i}$ becomes a realization of a scaled Brownian Motion (BM) process (also known as the Wiener stochastic process) with covariance matrix

$$H_i^{j_1 j_2} = \phi t_{j_1}$$

for $j_1, j_2 = 1, \ldots, n_i$ and $j_1 \leq j_2$ (Taylor et al. 1994). The BM stochastic process is parameterized by a single parameter ϕ and can be interpreted as no derivative tracking (Sy et al. 1997). When $\mathbf{w_i}$ is the realization of a scaled BM process we shall refer to model (1) as a linear mixed effects BM model. The covariance matrix of $\mathbf{y_i}$ is $\mathbf{V_i} = \mathbf{Z_i}\mathbf{G}\mathbf{Z_i^T} + \mathbf{H_i} + \sigma^2\mathbf{I_{n_i}}$ (Patterson and Thompson 1971) and we denote the vector of unknown variance parameters by θ, which consists of the unique components of $\mathbf{G}$, the IOU or BM parameters, and σ^2.

2.1 Fitting of the model

The model is fitted using restricted maximum likelihood (REML) estimation (Patterson and Thompson 1971). REML estimates of θ are calculated using an optimization algorithm: the Newton-Raphson (NR) algorithm, the Fisher Scoring (FS) algorithm, the Average Information (AI) algorithm or a combination of these (Gumedze and Dunne 2011). The FS and AI algorithms are variants of the NR algorithm. The FS algorithm replaces the observed information matrix by the expected information matrix in the NR algorithm, and the AI algorithm replaces the observed information matrix by the average of the observed and expected information matrices (called the average information matrix). The convergence time for the NR algorithm is quicker than the FS algorithm because the NR algorithm converges in fewer iterations and its cost-per-iteration is only slightly slower than that of the FS algorithm

(Gumedze and Dunne 2011). However, the FS algorithm is more robust to poor starting values than the NR algorithm and so Jennrich and Sampson (Jennrich and Sampson 1976) recommended starting with a few iterations of the FS algorithm and then switching to the NR algorithm.

We provide two methods for calculating starting values. The default method fits a standard linear mixed effects model using command `mixed`. The resulting estimates become the starting values for the random effects parameters and measurement error variance, whilst the IOU or BM parameters are set to fixed values representing strong derivative tracking ($\alpha = 1$ and $\tau = 0.1$ or $\phi = 0.01$). The alternative method derives all starting values from the data. First, the alternative method predicts the residuals of the response after accounting for the model's mean structure using commands `regress` and `predict`. Second, the data are discretized according to a given time-window interval, derived from the observed frequency of measurement. Third, it calculates the variance of the residuals within a time-window interval, and the covariance of the residuals between time-window intervals. Starting values are then calculated based on these variances and covariances, and their changes over time. For example, for a model with a random intercept and IOU process the linear change in residual variances over time gives an approximate estimate for the ratio $\omega = (\tau/\alpha)^2$.

Taylor et al. (1994) parameterized the IOU process as α and $\omega = (\tau/\alpha)^2$ and experienced convergence problems as α became increasingly large or small. Taylor et al. suggested reparameterizing α as $\ln\alpha$ or as α^{-2} if α was suspected to be large. We allow 6 different parameterizations of the IOU process: $[\alpha;\tau]$, $[\alpha;\omega]$, $[\ln\alpha;\tau]$, $[\ln\alpha;\omega]$, $[\alpha^{-2};\tau]$ and $[\alpha^{-2};\omega]$.

The restricted log-likelihood is profiled with respect to σ^2 to reduce the number of parameters to be optimized. The optimized parameters θ^* are the unique elements of the log-cholesky parameterization of $G^* = \sigma^{-2}G$, selected IOU parameterization (with τ/σ or ω/σ^2) or BM parameter ϕ/σ^2. The optimization algorithm finds the value of θ^* that minimizes the negative of twice the profiled restricted log-likelihood. Once minimization with respect to θ^* is completed, the REML estimates of (θ^*, σ^2) are transformed to parameters with ranges $(-\infty, \infty)$ and the information matrix with respect to these transformed parameters is calculated. Normal-based 95% confidence intervals are calculated and the end-points are back-transformed to the required scale (e.g., G, α, τ and σ^2). The variance-covariance matrix of the estimates on the untransformed scale is calculated using the delta method (Oehlert 1992; Rice 1994).

We implemented `xtmixediou` using Stata's matrix programming language Mata and used the inbuilt Mata function `optimize` to perform the optimization.

3 The xtmixediou command

3.1 Syntax

The `xtmixediou` command fits the linear mixed effects IOU model (or the linear mixed effects BM model), as described in section 2. The command assumes the data are in long form (see [D] **reshape**), and the command is compatible with Stata versions 11 and above. The syntax of `xtmixediou` is as follows:

`xtmixediou` *depvar* [*indepvars*] [*if*] [*in*] , `id(`*levelvar*`) time(`*timevar*`)`
[`nofeconstant reffects(`*varlist*`) noreconstant iou(`*ioutype*`) brownian`
`svdataderived algorithm(`*algorithm_spec*`) iterate(`#`) difficult nolog trace`
`gradient showstep hessian`]

depvar is the dependent variable Y_i, which contains the repeated measurements.

indepvars are the covariates X_i for the fixed portion of the model (i.e., the fixed effects). `xtmixediou` automatically includes a constant term (i.e., an intercept) in the fixed effects. Factor variables are allowed (see [U] **11.4.3 Factor variables**).

3.2 Options

`id`(*levelvar*) defines the variable for identifying individuals (i.e., the level-2 units). *levelvar* can be a numeric variable or a string variable. This is a required option.

`time`(*timevar*) defines the numeric variable for the timepoints $\mathbf{t_i}$ at which the measurements of *depvar* were observed. This is a required option.

`nofeconstant` suppresses the constant term for the fixed portion of the model. By default a constant term is included in the fixed portion of the model.

`reffects`(*varlist*) defines the random effects of the model. `xtmixediou` automatically includes a constant term in the random effects. For two or more random effects an unstructured covariance matrix is assumed (i.e., all variances and covariances are distinctly estimated). Factor variables are not allowed. The default (when `reffects`(*varlist*) is not specified) is a random intercept.

`noreconstant` suppresses the constant term for the random effects of the model. By default a constant term is included in the random portion of the model.

`iou`(*ioutype*) specifies the parameterization of the IOU process used during estimation, where *ioutype* is one of six parameterizations given in table 1. The default parameterization is α and τ. Changing the IOU parameterization may improve convergence. For example, parameterizations $\ln \alpha$ or α^{-2} may be useful if α is suspected to be large. There is no guarantee that the other parameterizations will work better than the default; sometimes it is better and sometimes it is worse.

Table 1: IOU parameterizations

ioutype	Description
`at`	α and τ, the default
`ao`	α and $\omega = (\tau \div \alpha)^2$
`lnat`	$\ln \alpha$ and τ
`lnao`	$\ln \alpha$ and $\omega = (\tau \div \alpha)^2$
`isat`	α^{-2} and τ
`isao`	α^{-2} and $\omega = (\tau \div \alpha)^2$

`brownian` specifies a scaled BM process, a special case of the IOU process (see section 2), which is parameterized by a single parameter ϕ. The BM process represents no derivative tracking and the fitted model then becomes the linear mixed effects BM model.

`svdataderived` specifies that the starting values of all of the model's variance parameters (i.e., random effects variances and covariances, IOU or BM parameters and measurement error variance) are derived from the data (see section 2.1). `svdataderived` assumes the user has specified (using options `reffects()` and/or `noreconstant`) that the random effects only include a random intercept and/or a random linear slope. When `svdataderived` is not specified (i.e., the default) then a linear mixed effects model without an added IOU or BM process is fitted (using Stata's command `mixed`) and the resulting expectation maximization estimates are used as the starting values for the random effects variances and covariances, and the measurement error variance; whilst the starting values for the IOU or BM parameters are set to small positive values (i.e., representing strong derivative tracking). `xtmixediou` saves the starting values to matrix `e(sv)`.

`algorithm`(*algorithm_spec*) where *algorithm_spec* is

algorithm [# [*algorithm* [#]] ...]

and *algorithm* is {`nr` | `fs` | `ai`}.

`algorithm(nr)` specifies the Newton-Raphson (NR) algorithm.

`algorithm(fs)` specifies the Fisher Scoring (FS) algorithm.

`algorithm(ai)` specifies the Average Information (AI) algorithm.

The default is `algorithm(nr)`.

The user can switch between algorithms by specifying more than one in the `algorithm()` option. If the number of iterations (#) is not specified then by default an algorithm is used for five iterations before switching to the next algorithm. To specify a different number of iterations, include the number after the algorithm's abbreviation in the option. For example, specifying `algorithm(fs 10 nr 100)` requests using 10 iterations using the FS algorithm followed by 100 iterations using the NR algorithm, and then switches back to FS for 10 iterations, and so on. The process continues until convergence or until the maximum number of iterations is

reached.

Convergence of the NR algorithm may be improved by starting with a few, say 3, iterations of the FS or AI algorithm, especially when the starting values of the parameters are suspected to be far from the REML estimates.

The following options are also described in [R] **maximize**.

`iterate`(#) specifies the maximum number of iterations. The optimizer stops and presents the current results when either the number of iterations equals `iterate`(#) or convergence is declared before the number of iterations equals the specified maximum. The default value is `iterate`(`16000`).

`difficult` specifies that the likelihood function is likely to be difficult to maximize because of nonconcave regions (i.e., when the message "not concave" appears repeatedly) and that the standard stepping algorithm is not working well. `difficult` specifies that a different stepping algorithm be used in the nonconcave regions. There is no guarantee that `difficult` will work better than the default; sometimes it is better and sometimes it is worse. You should use the `difficult` option only when the default stepper declares convergence and the last iteration is "not concave" or when the default stepper is repeatedly issuing "not concave" messages and producing only tiny improvements in the log likelihood.

`nolog` suppresses the display of the iteration log showing the progress of the log likelihood. The log is displayed by default.

`trace` adds to the iteration log a display of the current parameter vector.

`gradient` adds to the iteration log a display of the current gradient vector.

`showstep` adds to the iteration log a report on the steps within an iteration.

`hessian` adds to the iteration log a display of the current negative Hessian matrix.

3.3 Saved results

`xtmixediou` saves the following results to `e()`:

Scalars

`e(N)`	number of observations	`e(k)`	number of parameters
`e(k_f)`	number of fixed effects parameters	`e(k_r)`	number of random effects parameters
`e(k_res)`	number of residual-error parameters	`e(ll)`	restricted log-likelihood
`e(converged)`	1 if converged, 0 otherwise		

Macros

`e(cmd)`	`xtmixediou`	`e(cmdline)`	command as typed
`e(title)`	title in estimation output	`e(depvar)`	name of dependent variable
`e(id)`	name of variable identifying level-2 units	`e(time)`	name of timepoint variable for *depvar*
`e(revars)`	names of random effects variables	`e(redim)`	random effects dimension

`e(iou)`	`iou()` specification	`e(method)`	REML
`e(ml_method)`	type of ml method	`e(opt)`	type of optimization
`e(predict)`	command used to implement predict	`e(properties)`	b V

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators
`e(sv)`	starting values of the variance parameters	`e(N_g)`	group counts
`e(g_min)`	group-size minimum	`e(g_avg)`	group-size average
`e(g_max)`	group-size maximum		

Functions

`e(sample)`	marks estimation sample

3.4 Syntax for `predict`

Command `xtmixediou` supports the postestimation command `predict` (see [R] **predict**) to compute linear predictions, standard errors, fitted values and residuals. The syntax for `predict` following `xtmixediou` is

`predict` *newvarname* [*if*] [*in*] , [`xb stdp fitted residuals`]

statistic	Description
`xb`	linear prediction for the fixed portion of the model only; the default
`stdp`	standard error of the fixed portion linear prediction
`fitted`	fitted values, fixed portion linear prediction plus contributions based on predicted random effects and the realizations of the IOU (or BM) process
`residuals`	residuals, response minus fitted values

4 Example

The data for this example are simulated based on characteristics of data from a HIV/AIDS cohort study (UK CHIC study 2004). This study routinely collects clinical information on HIV-positive individuals aged over 16 years who have attended one of the collaborating centres for care at any time since 1996. One of the purposes of the study is to analyse the data to monitor response to antiretroviral therapy. A patient's repeated measurements of CD4 cell counts reflect both HIV disease progression and recovery after a patient starts therapy (Sabin and Lundgren 2013). For example, an analysis using a standard linear mixed effects model (i.e., Stata command `mixed`) showed that CD4 cell counts continue to increase up to 8 years after initiation of therapy among patients who maintained virological suppression (Hughes et al. 2011). The analysis fitted a strong derivative tracking model that assumed a patient's CD4 counts maintained (or closely tracked) the same parametric trajectory (a two-degree fractional polynomial (Royston and Altman 1994)) throughout the patient's follow-up, and that within-patient residuals were uncorrelated with constant variance over time. Taylor et al 1994 state that it is unlikely that something as complex as a measurement of a patient's immune system would maintain the same parametric trajectory over long periods

of time. In our original analysis we were interested in the population trajectories (i.e., fixed effects), which are robust to the assumption of strong derivative tracking and incorrect specification of the variance structure. However, such robustness may not apply when one is interested in patient-level predictions (Taylor and Law 1998).

In the following analysis, we fit a linear mixed effects IOU model, a linear mixed effects BM model and a standard linear mixed effects model and compare their model fit and accuracy of patient-level predictions.

4.1 The data

The original dataset consisted of data on 18045 patients, who were expected to attend a HIV clinic about every 3 months. These patients had not received previous treatment for HIV, started therapy after 1997, had at least one CD4 cell count measurement before start of therapy and at least two CD4 cell count measurements during follow-up. Also, these patients had recorded values for the following pre-therapy (or baseline) patient characteristics: sex, age at start of therapy, ethnicity (white, black African, other), risk group for HIV infection (homosexual, heterosexual, other) and pre-therapy CD4 cell count group (0 to 99, 100 to 199, 200 to 349 and $\geq$ 350 cells/mm^3).

We simulated an unbalanced dataset of 1000 patients in three separate stages. In the first stage patient characteristics were simulated under a general location model (Olkin and Tate 1961). In the second stage the number of measurements per patient, the length of follow-up and the time intervals between consecutive measurements within a patient were simulated based on these features of the original dataset. And in the third stage we simulated longitudinal CD4 counts (on the natural logarithm scale) under a linear mixed effects BM model, where the population trajectory was described by a fractional polynomial with powers 0 (interpreted as a natural log transformation) and 0.5, the aforementioned pre-therapy patient characteristics were also included as fixed effects, and the fractional polynomial power 0.5 and the intercept were included as random effects, with an unstructured random effects covariance matrix. The parameters of the models were set to the (restricted) maximum likelihood estimates from fitting the same models to the original dataset.

The following code (with corresponding output) describes the simulated dataset and lists the possible values of the categorical variables. Variable `patid` uniquely identifies a patient and variables `sex`, `age`, `ethnicity`, `risk`, and `baseline_cd4` are the pre-therapy characteristics. `cd4` is the CD4 cell count measurement (cells/mm^3) on its original scale and `lncd4` is its corresponding value on the natural logarithm scale. `time` is time in years of the CD4 cell count measurement since initiation of therapy. The data are in long format and the following code (with corresponding output) displays the first 3 measurements for 2 patients.

```
. use lncd4, clear
(example for xtmixediou)
. describe
Contains data from lncd4.dta
```

```
  obs:        15,526                          example for xtmixediou
 vars:            10                          14 Sep 2016 15:28
 size:       853,930
```

variable name	storage type	display format	value label	variable label
patid	int	%8.0g		Patient identifier
measurement	byte	%8.0g		Measurement occasion
time	float	%9.0g		Time of CD4 measurement since start of therapy (in years)
cd4	float	%9.0g		CD4 cell count measurement
lncd4	float	%9.0g		Natural logarithm CD4 count
sex	double	%9.0g	sexLabel	Sex
ethnicity	double	%15.0g	ethnicLabel	Ethnicity group
risk	double	%12.0g	riskLabel	Risk group for infection
baselinecd4	double	%10.0g	preCD4Label	baselineCD4_cat
age	double	%9.0g		Age at start of therapy

```
Sorted by:  patid  time
. label list sexLabel
sexLabel:
           0 male
           1 female
. label list ethnicLabel
ethnicLabel:
           0 white
           1 black African
           2 other ethnicity
. label list riskLabel
riskLabel:
           0 homosexual
           1 heterosexual
           2 other risk
. label list preCD4Label
preCD4Label:
           0 0 to 99
           1 100 to 199
           2 200 to 349
           3 350 plus

. format time lncd4 age  %4.2g
. list if (patid == 12 | patid==13) & time <=1, noobs sep(0) abbr(3) str(3) c
```

patid	mea~t	time	cd4	lncd4	sex	eth~y	risk	bas~4	age
12	1	.2	13	2.6	male	white	hom..	0 t..	25
12	2	.49	23	3.1	male	white	hom..	0 t..	25
12	3	.87	16	2.8	male	white	hom..	0 t..	25
13	1	.25	22	3.1	male	white	hom..	0 t..	29
13	2	.45	34	3.5	male	white	hom..	0 t..	29
13	3	.67	36	3.6	male	white	hom..	0 t..	29

4.2 Using command xtmixediou to fit a linear mixed effects IOU model

We shall fit a series of linear mixed effects models with different variance structures but the same mean structure, listed in table 2. For the fixed portion of all models we include the pre-therapy variables as time-independent covariates, and the population trajectory is modelled by a fractional polynomial with powers 0 and 0.5.

Table 2: Variance structures of the fitted models[♭]

Model	Random effects	Stochastic process
riiou	constant	Integrated Ornstein-Uhlenbeck (IOU)
ribm	constant	Brownian Motion (BM)
rfpiou	constant and $\texttt{time}^{0.5}$	IOU
rfpbm[♯]	constant and $\texttt{time}^{0.5}$	BM
ri	constant	-
rfp	constant, $\texttt{time}^{0.5}$ and ln(time)	-

[♭] All models include measurement error variance; [♯] Model used to simulate the data

First, we generate the fractional polynomial powers of `time`. We do not need to change the origin of `time` nor rescale the variable because all of its values are greater than 0 and its standard deviation is close to 1. (See [R] **fracpoly**).

```
. summarize time
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        time |      15526    2.374024    1.407925   .1149897   4.999316

. gen time_ln = ln(time)
. gen time_05 = time^0.5
```

We begin by fitting a linear mixed effects IOU model, where the only random effect is a random intercept, using the following code. We refer to this model as a random intercept IOU (`riiou`) model. The `xtmixediou` command supports the use of Stata's factor notation (see [U] **11.4.3 Factor variables**). The code below specifies that the reference categories for `ethnicity`, `risk`, and `baseline_cd4` are respectively white, homosexual, and 200 to 349 cells/mm^3. Following Stata's convention, by default an intercept is automatically added to the fixed effects and to the random effects. Therefore, as the model only contains a random intercept we do not need to specify option `reffects()`. The required options `id()` and `time()` respectively declare that variable `patid` is the unique identifier for patients and `time` contains the measurement times. We specify that all starting values are derived from the data using option `svdataderived`. Lastly, we store the estimation results to `riiou_model` and, predict the fitted values and residuals.

```
. xtmixediou lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4, id(patid) time(time) svdata
  (output omitted)
. estimates store riiou_model
```

```
. predict riiou_fit, fitted
. predict riiou_res, residuals
```

Below displays the output of the `xtmixediou` command. The layout of the output follows that of Stata's command `mixed`. The total number of observations and the minimum, maximum, and average number of observations per patient are displayed at the top right. Then follows a table displaying the results for the fixed effects, random effects, IOU or BM parameters, and the variance of the measurement error.

```
Linear mixed IOU REML regression          Number of obs       =         15526
                                          Number of groups    =          1000

                                          Obs per group : min =             2
                                                          avg =          15.5
Restricted log likelihood = -6169.4427                    max =            26
```

lncd4	Coef.	Std. Err.	z	P >|z|	[95% Conf.	Interval]
time_ln	.1232436	.0223509	5.51	0.000	.0794366	.1670506
time_05	.077378	.0500194	1.55	0.122	-.0206582	.1754142
age	-.0000926	.0014625	-0.06	0.950	-.002959	.0027738
sex	.0923211	.0441723	2.09	0.037	.0057449	.1788972
risk						
heterosexual	-.1314315	.0452229	-2.91	0.004	-.2200668	-.0427961
other risk	-.1403481	.0555603	-2.53	0.012	-.2492443	-.0314519
ethnicity						
black African	-.1117199	.0455415	-2.45	0.014	-.2009796	-.0224601
other ethnic~y	-.1119597	.0382533	-2.93	0.003	-.1869347	-.0369847
baselinecd4						
0 to 99	-1.216405	.0362109	-33.59	0.000	-1.287377	-1.145433
100 to 199	-.3562389	.0354835	-10.04	0.000	-.4257853	-.2866925
350 plus	.4131572	.0405326	10.19	0.000	.3337148	.4925996
_cons	4.151499	.0803116	51.69	0.000	3.994091	4.308907

Variance parameters	Estimate	Std. Err.	[95% Conf.	Interval]
Random-effects:				
Var(_cons)	.1320698	.0080314	.1172301	.148788
IOU-effects:				
alpha	.9403315	.1105896	.7467442	1.184105
tau	.4873562	.0409801	.4133049	.5746751
Var(Measure. Err.)	.0747382	.0011132	.0725879	.0769522

The top two fixed effects are the fractional polynomial powers, which describe the population average trajectory of `lncd4`. Fixed effect `_cons` describes the population average of `lncd4` at the start of therapy among white, homosexual males with pre-therapy CD4 cell count between 200 and 349 cells/mm^3, and aged 0 years. The remaining fixed effects describe population average differences in `lncd4`, at the start of therapy, between different patient groups. In the second table, `Var(_cons)` describes the between subject

variance (at start of therapy) after controlling for the fixed effects. The estimated value of α is quite small, indicating fairly strong derivative tracking. Lastly, `Var(Measure. Err.)` describes the variance of the measurement level errors (i.e., the residuals).

Below, we fit a random intercept BM model (`ribm`) by adding option `brownian` to the previous code (results not shown), store the estimation results to `ribm_model`, and make predictions under this model. The results of the linear mixed effects BM model has the same layout as above except within the table of variance parameters a single parameter `phi` replaces the IOU parameters `alpha` and `tau`.

```
. xtmixediou lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4, id(patid) time(time) svdata brownian
  (output omitted)
. estimates store ribm_model
. predict ribm_fit, fitted
. predict ribm_res, residuals
```

We can specify the fractional polynomial powers as random using option `reffects()` (demonstrated in the following code). We shall refer to a model with a random intercept, and one or more random fractional polynomial powers plus IOU or BM process as a random fractional polynomial IOU or BM model (`rfpiou` or `rfpbm`), respectively. Note, the data model used to simulate the data was the random fractional polynomial BM model. Because the random effects include variables that are neither a random intercept nor a random linear slope we cannot use option `svdataderived`. When we fitted a model with both fractional polynomial powers as random effects (with an IOU or BM process) the corresponding variances and covariances associated with power 0 were close to zero (results not shown). Therefore, we have excluded the random effect for power 0. For the random fractional polynomial IOU model we use option `difficult` because of nonconcave regions.

```
. * random fractional polynomial IOU model
. xtmixediou lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4, id(patid) time(time) ///
> reffects(time_05) difficult
  (output omitted)
. estimates store rfpiou_model
. predict rfpiou_fit, fitted
. predict rfpiou_res, residuals
.
. * random fractional polynomial BM model
. xtmixediou lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4, id(patid) time(time) ///
> reffects(time_05) brownian
  (output omitted)
. estimates store rfpbm_model
. predict rfpbm_fit, fitted
. predict rfpbm_res, residuals
```

For comparison we also fit two standard linear mixed effects model (i.e., without an

IOU or BM process) using command `mixed` (code shown below), where: (1) only the intercept is random (`ri`), and (2) the intercept and fractional polynomial powers 0 and 0.5 are included as random effects (`rfp`). For the latter model, none of the estimates for the random effects variances and covariances are close to 0, and a model that includes both powers as random effects has a lower deviance, Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values than a model that excludes power 0 as a random effect (results not shown). We save the data and the predictions to filename `lncd4_predictions`.

```
. * random intercept model
. mixed lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4 || patid: , var reml
  (output omitted)
. estimates store ri_model
. predict ri_fit, fitted
. predict ri_res, residuals
.
. * random fractional polynomial model
. mixed lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4 || patid: time_ln time_05, var reml cov(unstructured)
  (output omitted)
. estimates store rfp_model
. predict rfp_fit, fitted
. predict rfp_res, residuals
. save lncd4_predictions, replace
file lncd4_predictions.dta saved
```

We use Stata's command `estimates stats` to compare the models with respect to the AIC and BIC values. The AIC and BIC values for the random intercept model (`ri`) are almost double the corresponding values for the other models indicating that this model is by far the poorest fit to the data (see code and output below). The model with the lowest AIC and BIC values is the random fractional polynomial BM model (the model used to simulate the data), although the AIC and BIC values for the random fractional polynomial IOU model are very similar. Based on these criteria a user would select a model with an IOU or BM process over the random fractional polynomial model (without an IOU or BM process).

Command `estimates stats` calculates the AIC value as $-2\ln L + 2k$, and the BIC value as $-2\ln L + k \times \ln N$, where $\ln L$ is the maximized log-likelihood of the model, $k = p+q$ is the number of fixed effects coefficients (p) plus the number of variance parameters (q), and N is the sample size (see [R] **estat**). For REML estimation the AIC and BIC values can also be calculated as $-2\ln L+2q$ and $-2\ln L+\ln(N-p)\times q$ respectively (Smith 2011). The AIC and BIC values calculated using the latter formulae are very similar to those calculated by `estimates stats`, and lead to the same conclusions (results not shown). Note, the AIC and BIC values (of both sets of formulae) are based on the (restricted) log-likelihood of the marginal model $y \sim N(X\beta, V)$. Criteria based on the marginal model may not be reliable for selection of the variance structure of a linear mixed model (Vaida and Blanchard B iometrika; Liang et al. 2008; Greven and Kneib

2010; Müller et al. 2013).

```
. estimates stats riiou_model ribm_model rfpiou_model rfpbm_model ri_model ///
> rfp_model
```

Model	Obs	ll(null)	ll(model)	df	AIC	BIC
riiou_model	15526	.	-6169.443	16	12370.89	12493.29
ribm_model	15526	.	-6249.674	15	12529.35	12644.1
rfpiou_model	15526	.	-6046.815	18	12129.63	12267.34
rfpbm_model	15526	.	-6046.857	17	12127.71	12257.77
ri_model	15526	.	-11226.74	14	22481.47	22588.58
rfp_model	15526	.	-6377.38	19	12792.76	12938.11

```
Note:  N=Obs used in calculating BIC; see [R] BIC note
```

Lastly, among the linear mixed IOU and BM models (`riiou`, `ribm`, `rfpiou`, and `rfpbm`) the estimates of fixed effect $\ln(time)$ (fractional power 0) were slightly larger than those from the standard linear mixed effects models (`ri` and `rfp`), whilst estimates of fixed effect $time^{0.5}$ were slightly smaller among the linear mixed effects IOU and BM models. However, for both fractional power fixed effects, the 95% confidence intervals from all models overlapped. The estimates of the remaining fixed effects (baseline characteristics) were similar among all models. (Results for the fixed effects estimates not shown).

4.3 Comparison of the variance structures

The previous 6 models make different assumptions about how the variance of `lncd4` changes over time, and about how the correlation between measurements changes over time. For each model in turn, using its variance function and estimates of the variance parameters we can plot a model's assumed pattern of variances and correlations over time. To further assess model fit, we shall compare the models' patterns in variances and correlations with the observed changes in variance and correlation of `lncd4`. Section 6 shows how we derive variables for the observed variances of `lncd4` and the observed correlations with the first measurement, and the corresponding variances and correlations under the 6 models. The appendix also includes the code for generating figures 2 and 3.

Figure 2 shows the changes in variance over time, where the observed variances are displayed as scatter points and the model variance patterns as lines. The variance patterns of all models except the random intercept model (`ribm`) and the random intercept BM model (`ribm`) closely follow the observed changes in variance over time.

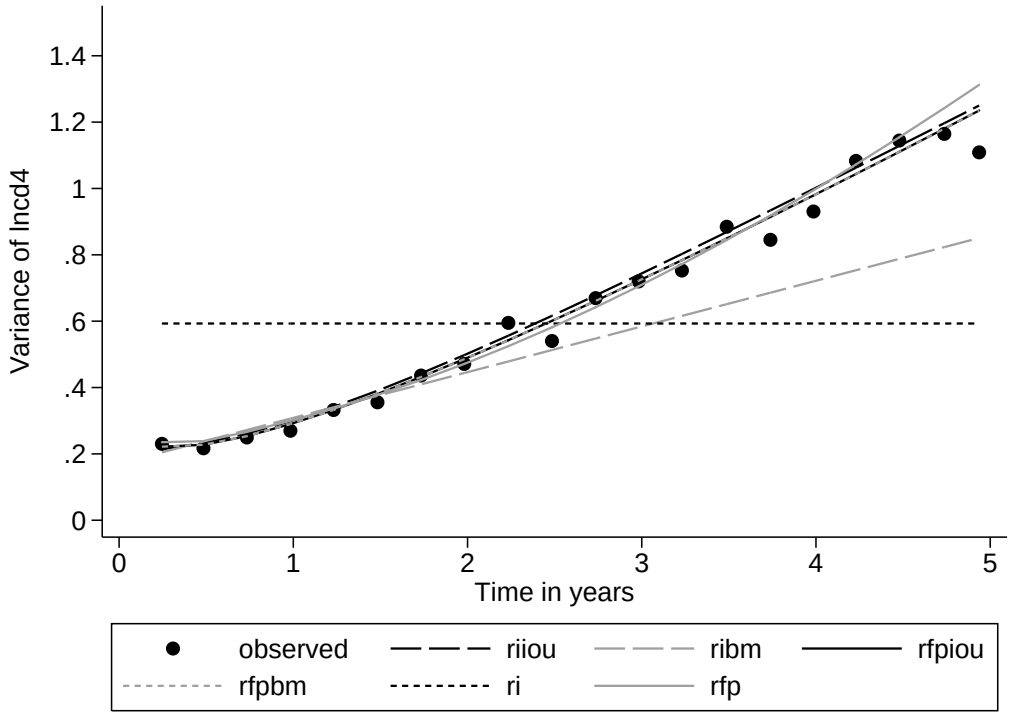

Figure 2: Changes in variance over time

Figure 3 shows the changes in the correlations (with the first measurement) over time, where the observed correlations are displayed as scatter points and the model correlation patterns as lines. The correlation patterns for the random fractional polynomial BM model (`rfpbm`) and the random fractional polynomial IOU model are virtually identical, with the two patterns overlaying each other. The correlation patterns that most closely follow the observed changes are for the three models that include at least one of the fractional polynomial powers as a random effect (`rfp`, `rfpiou`, and `rfpbm`). Given that model `rfp` does not include an added stochastic process, the similarity of the correlation pattern of model `rfp` with those of models `rfpiou` and `rfpbm` may be explained by the additional random effect (for power 0) present in model `rfp` which is not present in models `rfpiou` and `rfpbm` (see table 2). Note, we considered correlations with the first measurement as a reference because we could calculate at least one correlation for all subjects (i.e., the minimum number of measurements was 2), and similarly, we could have considered correlations with the second measurement as a reference. However, using the third or later measurements as a reference would have resulted in some subjects being excluded from the correlation calculations.

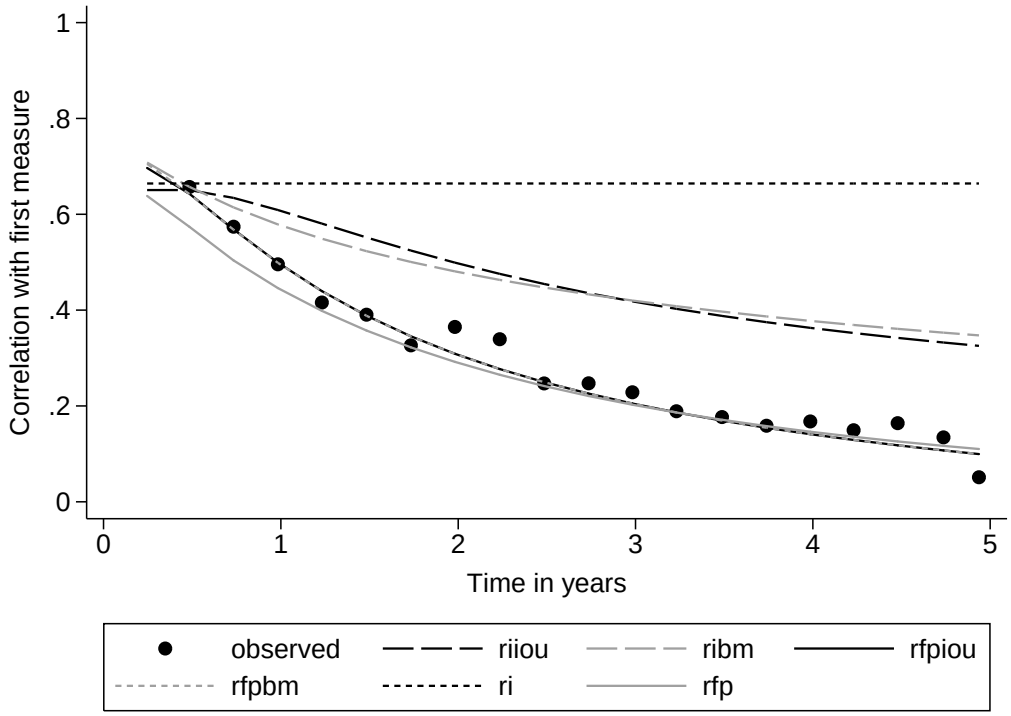

Figure 3: Changes in correlation over time

4.4 Comparison of the fitted values

In this section we compare the fitted values of the 6 models with respect to two measures: (1) the mean squared error (MSE), which is the average squared difference between the fitted values and the observed values, and (2) the percentage of fitted values within 5% of the observed values.

We have previously saved the fitted values of the 6 models to dataset `lncd4_predictions`. First, separately for each model, we generate a variable for the squared difference between the fitted and observed values, and a variable to indicate if a fitted value is within 5% of the observed value. We then use the `collapse` command to calculate the average squared difference across all measurements and to sum the number of fitted values within the 5% interval.

```
. use lncd4_predictions, clear
.
. * lower and upper limits for 5% interval
. gen ll5 = lncd4 - 0.05*lncd4
```

```
. gen ul5 = lncd4 + 0.05*lncd4
.
. local listing "riiou ribm rfpiou rfpbm ri rfp"
. foreach model of local listing {
  2.         * squared difference
.         gen mse_`model´ = (`model´_fit - lncd4)^2
  3.
.         * indicator of within 5% interval
.         gen in5_`model´ = 1 if `model´_fit>=ll5 & `model´_fit<=ul5
  4. }
  (output omitted)
. collapse (mean)mse* (sum)in5*
. list mse*, clean noobs abbreviate(14)
    mse_riiou   mse_ribm   mse_rfpiou   mse_rfpbm     mse_ri    mse_rfp
     .0597014   .0381547     .0491328    .0464631   .1867051   .0727348
. list in5*, clean noobs abbreviate(14)
    in5_riiou   in5_ribm   in5_rfpiou   in5_rfpbm   in5_ri   in5_rfp
         8844      10441         9522        9738     5970      8227
```

The lower the MSE and the larger the number of values within the 5% interval then the greater the accuracy of the fitted values. The models without an added IOU or BM process (`ri` and `rfp`) generated the least accurate fitted values. The model that generated the fitted values closest to the observed values is the random intercept BM model, even though the data were simulated under the random fractional polynomial BM model, and model fit statistics, and figures 2 and 3 suggested that other models provided a better fit to the data. This is in keeping with previous findings that when fitting a linear mixed effects IOU model it is sufficient to include a random intercept plus the IOU or BM process, and predictions under the linear mixed effects IOU or BM model are robust to incorrect specification of the true covariance structure (Taylor et al. 1994; Taylor and Law 1998). If we are evaluating a model for its predictive ability then selection based on accuracy of prediction may be preferable. Also, as noted earlier, selecting the variance structure of a linear mixed effects model based on marginal model criteria may be unreliable.

5 Discussion

We have presented the new Stata command `xtmixediou`, which implements the linear mixed effects IOU model, and its special case the linear mixed effects BM model. The model allows for autocorrelation, changing within-subject variance, and the incorporation of derivative tracking; that is, how much a subject tends to maintain the same trajectory for extended periods of time. The data may be unbalanced with a differing number of measures per subject, and the time interval between consecutive measurements may differ within and between subjects. To make our command user friendly, we designed `xtmixediou` to have many of the same features as Stata's own regression commands. For example, the displayed results of `xtmixediou` follow the same format as that of Stata's `mixed` command and factor notation is supported. In situations where convergence problems occur the command allows the user to change the method for

deriving the starting values for optimization, the optimization algorithm and the parameterization of the IOU process. Also, we have incorporated Stata's `maximize` option `difficult`, which specifies using a different stepping algorithm in nonconcave regions. We also provide a `predict` command to generate predictions under the linear mixed effects IOU model.

A limitation of our `predict` command is that we do not provide best linear unbiased predictions (BLUPs) of the random effects nor realizations of the IOU (or BM) process. Solving Henderson's mixed model equations (Gumedze and Dunne 2011) for three unknowns (the fixed effects coefficients, random effects, and realizations of the stochastic process) entails complex matrix algebra. Instead we have solved these equations for two unknowns, the fixed effects coefficients and the random effects plus the realizations of the stochastic process, and so we are able to predict fitted values. In future work we will provide separate predictions for the random effects and the realizations of the stochastic process.

We are not aware of any other publicly available software that fits the linear mixed effects IOU model. We hope our command `xtmixediou` will encourage and help statisticians to apply the linear mixed effects IOU model to their data.

6 Appendix

We wish to examine the changes in the variance of `lncd4` over time after accounting for the mean structure of the model (which is the same for all six models). Therefore, we fit a linear regression model with the same mean structure and predict residuals under this model. We shall use these residuals to examine the variance structure of the observed data.

```
. use lncd4_predictions, clear
. regress lncd4 time_ln time_05 age sex i.risk i.ethnicity ///
> ib2.baselinecd4
  (output omitted)
. predict reg_res, residuals
```

Next we group the data according to the nearest 3 month interval and drop any duplicates where a patient has more than one measurement within the same 3 month interval. We then reshape the data into wide format and, for each interval, calculate the variance of the residuals and its correlation with the first measurement. During the calculation process the variances and correlations are stored in matrix `obs`, and afterwards the columns of the resulting matrix are converted into variables.

```
. * round to nearest 3-month interval
. gen record = round(time/0.25)
.
. * drop duplicate patient records within same interval
. duplicates drop patid record, force
.
. * maximum number of records per patient
```

```
. summarize record
  (output omitted)
.
. * reshape the data into wide format
. keep patid record reg_res time
. reshape wide reg_res time, i(patid) j(record)
(note: j = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20)
  (output omitted)
.
. * calculate variances and correlations across patients
. matrix obs = J(`max´,3,0)
. forvalues t=1(1)`max´ {
  2.         quietly summarize reg_res`t´, detail
  3.         matrix obs[`t´,1] = r(Var)
  4.         quietly summarize time`t´, detail
  5.         matrix obs[`t´,2] = r(mean)
  6.         quietly correlate reg_res1 reg_res`t´
  7.         matrix obs[`t´,3] = r(rho)
  8. }
. matrix obs[1,3] = .
.
. * create variables from matrix
. clear
. svmat obs
. rename obs1 obsvar
. rename obs2 obstime
. rename obs3 obscorr
```

Using the models' variance functions and parameters' estimates we generate the corresponding variances and correlations under the six models. We saved these data to filename `patterns`.

```
. * generate fractional polynomial powers
. gen obstime_ln = ln(obstime)
. gen obstime_05 = obstime^0.5
.
. * extract first timepoint
. local t1 = obstime in 1
. local t1_ln = ln(`t1´)
. local t1_05 = sqrt(`t1´)
.
. * random intercept IOU model
. scalar varRI = .1320698
. scalar alpha = .9403315
. scalar tau = .4873562
. scalar varME = .0747382
.         * variance over time
. gen riiou_var = varRI + ((tau^2)/(alpha^3))*(alpha*obstime + ///
>         exp(-alpha*obstime) - 1) + varME
.         * correlation with first measurement over time
. local t1 = obstime in 1
```

```
. local var1 = riiou_var in 1
. gen riiou_cov = varRI + ((tau^2)/(2*alpha^3))*(2*alpha*`t1´ + ///
>         exp(-alpha*`t1´) + exp(-alpha*obstime) - 1 ///
>                 - exp(-alpha*(obstime-`t1´)))
. gen riiou_corr = riiou_cov/(sqrt(`var1´)*sqrt(riiou_var))
.
. * random intercept BM model
. scalar varRI = .1110791
. scalar phi = .1377509
. scalar varME = .0597721
.         * variance over time
. gen ribm_var = varRI + phi*obstime + varME
.         * correlation with first measurement over time
. local var1 = ribm_var in 1
. gen ribm_cov = varRI + phi*`t1´
. gen ribm_corr = ribm_cov/(sqrt(`var1´)*sqrt(ribm_var))
.
. * random fractional polynomial IOU model
. scalar varR05 = .2872198
. scalar varRI =  .2699737
. scalar covRI05 =  -.2028851
. scalar alpha =  18.36982
. scalar tau = 5.134438
. scalar varME = .0672206
.         * variance over time
. gen rfpiou_var = varRI + varR05*obstime_05^2 + 2*covRI05*obstime_05 + ///
>         ((tau^2)/(alpha^3))*(alpha*obstime + exp(-alpha*obstime) -1) + varME
.         * correlation with first measurement over time
. local var1 = rfpiou_var in 1
. gen rfpiou_cov = varRI + varR05*`t1_05´*obstime_05 + ///
>         (`t1_05´ + obstime_05)*covRI05 + ///
>         ((tau^2)/(2*alpha^3))*(2*alpha*`t1´ + exp(-alpha*`t1´) + ///
>                 exp(-alpha*obstime) - 1 - exp(-alpha*(obstime-`t1´)))
. gen rfpiou_corr = rfpiou_cov/(sqrt(`var1´)*sqrt(rfpiou_var))
.
. * random fractional polynomial BM model
. scalar varR05 = .2881752
. scalar varRI = .2680412
. scalar covRI05 = -.2032494
. scalar phi = .0773855
. scalar varME =  .0653691
.         * variance over time
. gen rfpbm_var = varRI + varR05*obstime_05^2 + 2*covRI05*obstime_05 + ///
>         phi*obstime + varME
.         * correlation with first measurement over time
. local var1 = rfpbm_var in 1
. gen rfpbm_cov = varRI + varR05*`t1_05´*obstime_05 + ///
>         (`t1_05´ + obstime_05)*covRI05 + phi*`t1´
. gen rfpbm_corr = rfpbm_cov/(sqrt(`var1´)*sqrt(rfpbm_var))
.
```

```
. * random intercept model
. scalar varRI = .3939691
. scalar varME = .199089
.         * variance over time
. gen ri_var = varRI + varME
.         * correlation with first measurement over time
. local var1 = ri_var in 1
. gen ri_corr = varRI/(sqrt(`var1´)*sqrt(ri_var))
.
. * random fractional polynomial model
. scalar varRln = .2203064
. scalar varR05 =  1.329548
. scalar varRI =  1.538193
. scalar covln05 = -.4527635
. scalar covRIln = .5217772
. scalar covRI05 = -1.325447
. scalar varME =  .0850865
.         * variance over time
. gen rfp_var = varRI + varRln*obstime_ln^2 + varR05*obstime_05^2 + ///
>         2*covRIln*obstime_ln + 2*covRI05*obstime_05 + ///
>         2*covln05*obstime_ln*obstime_05 + varME
.         * correlation with first measurement over time
. gen rfp_cov = varRI + varRln*`t1_ln´*obstime_ln + ///
>         varR05*`t1_05´*obstime_05 + (`t1_ln´ + obstime_ln)*covRIln ///
>         + (`t1_05´ + obstime_05)*covRI05 + ///
>         (`t1_ln´*obstime_05 + obstime_ln*`t1_05´)*covln05
. local var1 = rfp_var in 1
. gen rfp_corr = rfp_cov/(sqrt(`var1´)*sqrt(rfp_var))
. save patterns, replace
file patterns.dta saved
```

Below is the Stata code for generating figures 2 and 3.

```
. * figure 2
. use patterns, clear
. scatter obsvar obstime, legend(label(1 "observed")) mcolor(gs0) || ///
> line riiou_var obstime, legend(label(2 "riiou")) ///
>         lcolor(gs0) lpattern(longdash) || ///
> line ribm_var obstime, legend(label(3 "ribm")) ///
>         lcolor(gs10) lpattern(longdash) || ///
> line rfpiou_var obstime, legend(label(4 "rfpiou")) ///
>         lcolor(gs0) lpattern(solid) || ///
> line rfpbm_var obstime, legend(label(5 "rfpbm") cols(4)) ///
>         lcolor(gs10) lpattern(shortdash) || ///
> line ri_var obstime, legend(label(6 "ri")) ///
>         lcolor(gs0) lpattern(shortdash) || ///
> line rfp_var obstime, legend(label(7 "rfp")) ///
>         lcolor(gs10) lpattern(solid) ///
> xtitle("Time in years") ytitle("Variance of lncd4") ///
> ylabel(0(0.2)1.5,angle(0)) plotregion(style(none))
.
. * figure 3
. scatter obscorr obstime, legend(label(1 "observed")) mcolor(gs0) || ///
> line riiou_corr obstime, legend(label(2 "riiou")) ///
```

```
>         lcolor(gs0) lpattern(longdash) || ///
> line ribm_corr obstime, legend(label(3 "ribm")) ///
>         lcolor(gs10) lpattern(longdash) || ///
> line rfpiou_corr obstime, legend(label(4 "rfpiou")) ///
>         lcolor(gs0) lpattern(solid) || ///
> line rfpbm_corr obstime, legend(label(5 "rfpbm") cols(4)) ///
>         lcolor(gs10) lpattern(shortdash) || ///
> line ri_corr obstime, legend(label(6 "ri")) ///
>         lcolor(gs0) lpattern(shortdash) || ///
> line rfp_corr obstime, legend(label(7 "rfp")) ///
>         lcolor(gs10) lpattern(solid) ///
> xtitle("Time in years",margin(small)) ylabel(0(0.2)1,angle(0)) ///
> ytitle("Correlation with first measure") plotregion(style(none))
```

7 Acknowledgements

We thank Professor Caroline Sabin, from the Research Department of Infection and Population Health, University College London, for granting access to the UK CHIC data. Rachael Hughes was supported by the Medical Research Council [grant number MR/J013773/1]. Jonathan Sterne was supported by grant number MR/J002380/1: this award was jointly funded by the UK Medical Research Council (MRC) and the UK Department for International Development (DFID) under the MRC/DFID Concordat agreement and is also part of the EDCTP2 programme supported by the European Union. He was also supported by National Institute for Health Research Senior Investigator award NF-SI-0611-10168. Kate Tilling was supported by the Medical Research Council Integrative Epidemiology Unit at the University of Bristol ($MC\_UU\_12013/9$).

8 References

Boscardin, W., J. Taylor, and N. Law. 1998. Longitudinal Models for AIDS Marker Data. *Stat Methods Med Res* 7: 13–27.

Greven, S., and T. Kneib. 2010. On the behaviour of marginal and conditional Akaike Information Criteria in linear mixed models. *Biometrika* 97: 773–789.

Gumedze, F., and T. Dunne. 2011. Parameter estimation and inference in the linear mixed model. *Linear Algebra Appl* 435: 1920–1944.

Hughes, R., M. Kenward, J. Sterne, and K. Tilling. 2017. Estimation of the linear mixed integrated Ornstein Uhlenbeck model. *Journal of Statistical Computation and Simulation* .

Hughes, R., J. Sterne, J. Walsh, L. Bansi, R. Gilson, C. Orkin, T. Hill, J. Ainsworth, J. Anderson, M. Gompels, D. Dunn, M. Johnson, A. Phillips, D. Pillay, C. Leen, P. Easterbrook, B. Gazzard, M. Fisher, and C. Sabin. 2011. Long-term trends in CD4 cell counts and impact of viral failure in individuals starting antiretroviral therapy: UK Collaborative HIV Cohort (CHIC) study. *HIV Medicine* 12: 583–593.

Jennrich, R., and P. Sampson. 1976. Newton-Raphson and Related Algorithms for Maximum Likelihood Variance Component Estimation. *Technometrics* 18: 11–17.

Laird, N., and J. Ware. 1982. Random-Effects Models for Longitudinal Data. *Biometrics* 38: 963–974.

Liang, H., H. Wu, and G. Zou. 2008. A note on conditional AIC for linear mixed-effects models. *Biometrika* 95: 773–778.

Müller, S., J. Scealy, and A. Welsh. 2013. Model Selection in Linear Mixed Models. *Statistical Science* 28: 135–167.

Oehlert, G. 1992. A Note on the Delta Method. *Am Stat* 46: 27–29.

Olkin, I., and R. Tate. 1961. Multivariate correlation models with mixed discrete and continuous variables. *The Annals of Mathematical Statistics* 32: 448–465.

Patterson, H., and R. Thompson. 1971. Recovery of inter-block information when block sizes are unequal. *Biometrika* 58: 545–554.

Rice, J. 1994. *Mathematical Statistics and Data Analysis*. 2nd ed. Duxbury.

Royston, P., and D. Altman. 1994. Regression Using Fractional Polynomials of Continuous Covariates: Parsimonious Parametric Modelling. *Applied Statistics* 43: 429–467.

Sabin, C., and J. Lundgren. 2013. The natural history of HIV infection. *Current Opinion in HIV and AIDS* 8: 311–317.

Smith, R. 2011. *Multilevel modeling of Social Problems, A Causal Perspective*. Springer New York.

Sowers, M. F., M. Crutchfield, J. F. Randolph, B. Shapiro, B. Zhang, M. L. Pietra, and M. A. Schork. 1998. Urinary Ovarian and Gonadotrophin Hormone Levels in Premenopausal Women With Low Bone Mass. *J Bone Miner Res* 13: 1191–1202.

Sy, J., J. Taylor, and W. Cumberland. 1997. A Stochastic Model for the Analysis of Bivariate Longitudinal AIDS Data. *Biometrics* 53: 542–555.

Taylor, J., W. Cumberland, and J. Sy. 1994. A Stochastic Model for Analysis of Longitudinal AIDS Data. *J Am Stat Assoc* 89: 727–736.

Taylor, J., and N. Law. 1998. Does the Covariance Structure Matter in Longitudinal Modelling for the Prediction of Future CD4 Counts? *Stat Med* 17: 2381–2394.

UK Collaborative HIV Cohort Steering Committee, . 2004. The creation of a large UK based multicentre cohort of HIV-infected individuals: the UK Collaborative HIV Cohort (UK CHIC) Study. *HIV Medicine* 5: 115–124.

Vaida, F., and S. Blanchard. B iometrika. Conditional Akaike information for mixed-effects models. *2005* 92: 351–370.

About the authors

Rachael A. Hughes is a Research Fellow at the School of Social and Community Medicine, University of Bristol. Her research interests focus on analysis in the presence of incomplete data, longitudinal analysis using mixed effects models and clinical epidemiology of HIV and AIDS in the era of antiretroviral therapy.

Michael G. Kenward has recently retired as Professor of Biostatistics at the London School of Hygiene and Tropical Medicine. His main interests are in handling of missing data, longitudinal data analysis, the design and analysis of cross-over trials and REML. He has co-authored texts on missing data in clinical research, multiple imputation and cross-over trials.

Jonathan A.C. Sterne is a Professor of Medical Statistics and Epidemiology, and is the head of the School of Social and Community Medicine at the University of Bristol. His main research interests are clinical epidemiology of HIV and AIDS in the era of anti-retroviral therapy; meta-analysis and systematic reviews; causal inference; methodology for epidemiology and health services research, epidemiology of asthma and allergic diseases.

Kate Tilling is a Professor of Medical Statistics at the School of Social and Community Medicine, University of Bristol. Her main interests are in longitudinal models, particularly multilevel models, and in methods to minimise the bias due to missing data. Applied interests include childhood growth, disease monitoring and progression, and modelling longitudinal changes in biomarkers.